

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-146842

(43)Date of publication of application : 06.06.1997

(51)Int.Cl.

G06F 12/08

G06F 12/08

G06F 3/06

G06F 3/06

(21)Application number : 07-300967

(71)Applicant : HITACHI LTD

(22)Date of filing : 20.11.1995

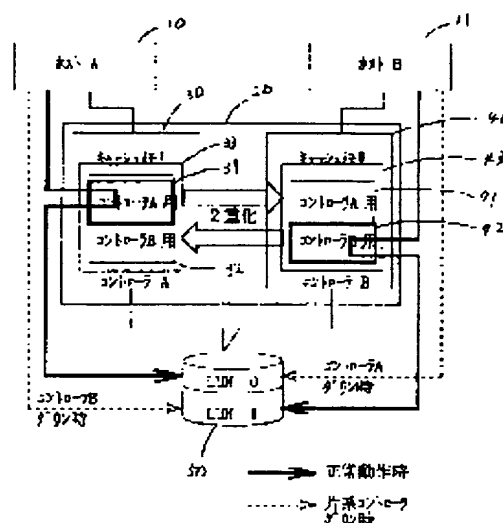
(72)Inventor : KOBAYASHI RIE  
MATSUMOTO YOSHIKO  
MURAOKA KENJI

## (54) STORAGE SUBSYSTEM

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide the storage subsystem with a controller which eliminates exclusive control over a cache between plural controllers sharing the cache.

**SOLUTION:** The cache areas of caches 33 and 43 in which data are written mutually in multiple are divided by processors and the controllers 30 and 40 access only their controller control areas. The cache areas that the controllers use are fixed to eliminate the need for exclusive control between the processors and prevent deterioration in performance due to multiprocessor constitution.



## LEGAL STATUS

[Date of request for examination]

31.10.2002

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's  
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

要登請求 未請求 解決項の数19 OL (全 14 頁)

(54) 【発明の名称】 配管サブシステム

2003 04 24 10.53

特開平 9 - 1 4 6 8 4 2

転送を制御するコントローラ3000、4000及びホストコンピュータ1200、1300とディスク装置7100間のデータ転送を制御するコントローラ500

【0023】コントローラ3000は、ホストコンピュータ1000とのプロトコル制御を行うホスト1、下制御部3100、コントローラ全体を制御するマイクロプロセッサ（以下「プロセッサ」という。）3200、データの伝送を管理するデータ転送制御部3300、ホストコンピュータ1000とディスク装置7000のデータ転送時及びプロセッサ間通信時に用いられるキャッシュ3400、各ディスク装置7000とのプロトコル制御を行うR/V1、下制御部3500より構成される。コントローラ4000、5000、6000はコントローラ3000と同一の構成である。

【0024】プロセッサ3200は、後述の手段により、あらかじめプロセッサ毎に静的に割り当てられた論理ボリュームの処理を行う。このプロセッサ毎の相対論理ボリュームの指定は、ホストコンピュータから論理ボリューム毎の相対プロセッサ指定コマンドを受け取ることにより、ダイナミックに設定可能である。このプロセッサと相対論理ボリュームとの対応情報は、後述のキャッシュ上の共通メモリ領域3410、4410に格納する。

【0025】データ転送制御部3300はプロセッサ3200からの指示により、ホストコンピュータ1000からのライトデータを指定キャッシュに多重書きする機能を備えている。この実施例の構成では、キャッシュ3400とキャッシュ4400の間で2重書きを行い、また、キャッシュ5400とキャッシュ6400の間で2重書きを行う。以下、キャッシュ3400とキャッシュ4400の2重に2重書きする方法について説明する。

【0026】キャッシュ3400とキャッシュ4400の内容について図3を用いて説明する。尚キャッシュ3400とキャッシュ4400は内部構成が同一であるため、キャッシュ3400を例に説明する。キャッシュ3400は、プロセッサ間通信に用いる制御情報を格納している共通メモリ領域3410、プロセッサ3200用領域3480、プロセッサ4200用領域3490より構成される。

【0027】プロセッサ3200用領域3480は、ホストコンピュータとディスク装置間のデータ転送時、データを1次的に格納するデータ格納エリア3482、データ格納エリア3482を管理するデータ管理情報3481より構成される。データ格納エリア3482に格納するライトデータと、このライトデータの管理情報は、キャッシュ4400内のプロセッサ3200用領域4480に2重書きを行う。同時に、プロセッサ4200用領域3490は、プロセッサ4200により、キャッシュ4400内のプロセッサ4200用領域4490のライトデータとライトデータの管理情報が2重書きされている。

【0028】共通メモリ領域3410は、論理ボリューム

に相当するプロセッサ情報3420、プロセッサ負荷情報3430、多重書き情報3450、プロセッサ間コミュニケーションシナメタ3460より構成される。これらの情報は、データ転送制御部3300、4300によって、キャッシュ3400と4400に2重書きされている。

【0029】図3（c）にプロセッサ間コミュニケーションシナメタの構成を示す。プロセッサ間コミュニケーションシナメタ3460は、プロセッサ3200、4200、5200、6200毎の書き込み用メモリ3461、3462、3463、3464より構成される。図3（d）にプロセッサ書き込み用メモリの構成を示す。プロセッサ3200書き込み用メモリ3461は、プロセッサ3200以外でプロセッサ4200、5200、6200への要求用エリア3471、3472、3473とプロセッサ3200以外のプロセッサ4200、5200、6200からの要求に対する応答用エリア3474、3475、3476より構成される。プロセッサ4200、5200、6200書き込み用メモリ3462、3463、3464の内部構成は、プロセッサ3200書き込み用メモリ3461と同一構成である。

【0030】キャッシュ5400とキャッシュ6400との間も、共通メモリ領域を除いて、キャッシュ3400とキャッシュ4400との間と同様に2重化が行われている。共通メモリ領域は、キャッシュ4000、4400に2重書きされている情報を制御装置内の全プロセッサで共有するため、キャッシュ5400、6400には存在しない。

【0031】本発明を実施する制御装置では、コントローラの増設はコントローラ2台単位で行い、残ったコントローラのキャッシュ間のみで2重書きを行うとともに、ドライブ毎のデータバスに、それぞれ異なるディスク装置は対になったコントローラとのみ接続することによりハードウェア構成を簡略化し、ドライブ側データバスとの接続を回避することが可能となる。

【0032】次に本実施例における、図2のディスクシステムでの、ホストコンピュータ1000からの1、〇処理について図4、図5、図6を用いて説明する。まず最初に、プロセッサ3200相当論理ボリュームへの1、〇処理について説明する。

【0033】図4は、ホストからの1、〇処理を示すフローチャートである。ホストコンピュータ1000からの書き込み要求時、プロセッサ3200は、まず、共通メモリ領域3410内の論理ボリューム相当プロセッサ情報3420によって、処理要求論理ボリュームの相当プロセッサ情報を取得し、自処理相当論理ボリューム（LUN）への処理の判定を行い（ステップ902）、自プロセッサ処理相当論理ボリュームへの処理であることを認識する。次に、処理後の判定を行い（ステップ903）、書き込み処理であることを認識する。

- 5 - 2003 04 24 10 53

図5を（ステップ911）。

【0042】図6において、プロセッサ3200からの要求を認識（ステップ931）したプロセッサ4200は、共通メモリ領域内の情報により、読み込み処理であることを認識する（ステップ932）、そして、共通メモリ領域から読み込み要求論理アドレス、読み込みデータのキャッシュ上の格納先アドレス、データ長を取得する（ステップ936）、次に、データをディスク装置7000からプロセッサ4200用領域4490に格納し、このデータをキャッシュ3400上の格納先アドレスに格納する（ステップ937）。さらに、共通メモリ領域3410、4410内のプロセッサ4200書き込み用メモリ内のプロセッサ3200からの要求に対する応答用エリアに読了情報を設定することにより、プロセッサ3200に読み込み終了を通知する（ステップ935）。

【0043】図4において、プロセッサ4200の処理を監視（ステップ912）していたプロセッサ3200は、この読み込み終了報告を受けて、データをホストコンピュータに転送する（ステップ913）。

【0044】このように、プロセッサ3200は、通常、プロセッサ3200用領域3480と4480を用いて、1、〇処理を行う。同時に、プロセッサ4200は、通常、プロセッサ4200用領域3490と4490を用いて、1、〇処理を行う。

【0045】このように、プロセッサ毎に使用するキャッシュ領域を固定化することにより、プロセッサ間の情報制御を簡便く、プロセッサ負荷増加に伴う性能劣化を防ぐことができる。特にホストコンピュータ間でファイル（論理ボリューム）をシェアしないシステムにおいては、接続しているコントローラ内のプロセッサとの論理ボリュームを割り当てておくことにより、1、〇処理の性能向上を可能とする。

【0046】次にコントローラ4000の動作時の（自動）の動作、復旧方式について図7、図8を用いて説明する。1、〇処理実行中、コントローラ4000の動作を監視したプロセッサは、共通メモリ領域3410を用いて、残りの全プロセッサに復旧開始を伝達する（ステップ972）。プロセッサ3200、5200、6200は、この復旧開始の伝達を受けて（ステップ955）、それぞれコントローラ4000のデータ転送制御部にキャッシュ3400と4400への2重書きを指示すると共に、共通メモリ領域3410、4410を用いて、処理後の応答をプロセッサ4200に伝達する（ステップ956）。この読了報告をプロセッサ4200から受信（ステップ973）したプロセッサ4200は、キャッシュ3400のデータ回復を行う（ステップ974）。データ回復が完了すると、共通メモリ領域3410、4410を用いて、プロセッサ3200に復旧完了を伝達する（ステップ975）。

【0047】図7において、この応答通知を受けた（ステップ958）プロセッサ3200は、プロセッサ4200用領域の初期状態をプロセッサ4200に復旧（ステップ959）させ、共通メモリ領域を用いて、初期状態の復旧をプロセッサ4200に伝達する（ステップ960）。図8において、この伝達を受けた（ステップ976）プロセッサ4200は、1、〇処理を再開する（ステップ977）。

【0048】図8は、障害が発生したコントローラ4000の復旧処理を示すフローチャートである。コントローラ4000の障害復旧が完了（ステップ971）されると、プロセッサ4200は、共通メモリ領域3410のデータ回復を行う（ステップ974）。データ回復が完了すると、共通メモリ領域3410、4410を用いて、プロセッサ3200に復旧完了を伝達する（ステップ975）。

【0049】図7において、この応答通知を受けた（ステップ958）プロセッサ3200は、プロセッサ4200用領域の初期状態をプロセッサ4200に復旧（ステップ959）させ、共通メモリ領域を用いて、初期状態の復旧をプロセッサ4200に伝達する（ステップ960）。図8において、この伝達を受けた（ステップ976）プロセッサ4200は、1、〇処理を再開する（ステップ977）。

【0050】尚、以上の実施例においては、コントローラ毎にプロセッサ、ホスト1、下制御部を1つ持った例を示したが、これらの数は任意である。ホストコンピュータからのコマンドを受け取ったプロセッサが、相対プロセッサに処理要求を伝達することにより、同時に処理できる。

【0051】また、キャッシュの分割方法は、プロセッサ毎に異なるが、ユーザの指定により任意に、変更可能である。特に、特定プロセッサをホットスタンバイ動作させる場合には、キャッシュに領域をホットスタンバイ

- 7 - 2003 04 24 10 53

ホスト1、下制御部3100により、書き込み論理ボリュームを受信し、データ転送制御部3300によってキャッシュ3400内のコントローラ3000用領域3480とキャッシュ4400のコントローラ3000用領域4480とに2重書きの管理情報とともに2重に格納する（ステップ904）。そして、この時点でホストコンピュータ1000に読了報告をする（ステップ905）。

【0034】例は、キャッシュ内のデータをディスク装置に格納する処理を示すフローチャートである。プロセッサ3200は、ホストコンピュータ1000からの1、〇処理とは異同期にプロセッサ3200用領域3480上のライトデータをデータ転送制御部3300とDR/V1、下制御部3500によりディスク装置7000に格納する（ステップ922）。この際、キャッシュメモリ領域により読み込みエラーが発生した場合（ステップ923）は、2重化しているプロセッサ3200用領域4480からディスク装置7000へ格納する（ステップ924）ことによりデータ損失を防止することができる。

【0035】ホストコンピュータ1000からの読み込み要求時、プロセッサ3200は、上記書き込み処理時、自プロセッサ処理相当論理ボリューム（LUN）への処理であることを認識（ステップ902）した後、処理後の判定を行う（ステップ903）、1、〇処理が読み込み処理であることを認識すると、データ転送制御部3300とDR/V1、下制御部3500によりデータをディスク装置7000からキャッシュ3400のコントローラ3000用領域3480に格納し（ステップ907）、ホストコンピュータに転送する（ステップ907）。

【0036】次にホストコンピュータ1000からの読み込み要求時、プロセッサ3200は、上記書き込み処理時、自プロセッサ処理相当論理ボリューム（LUN）への処理であることを認識（ステップ902）した後、処理後の判定を行う（ステップ903）、1、〇処理が読み込み処理であることを認識すると、データ転送制御部3300とDR/V1、下制御部3500によりデータをディスク装置7000からキャッシュ3400のコントローラ3000用領域3480に格納し（ステップ907）、ホストコンピュータに転送する（ステップ907）。

【0037】ホストコンピュータ1000からの書き込み要求時、プロセッサ3200は、まず、共通メモリ領域3410内の論理ボリューム相当プロセッサ情報3420によって、処理要求論理ボリュームの相当プロセッサ情報を取得し、自処理相当論理ボリュームへの処理の判定を行い（ステップ902）、処理相当論理ボリュームへの処理であることを認識する。次に、処理後の判定を行い（ステップ903）、書き込み処理であることを認識する。そして、ホストコンピュータ1000からの書き込み要求データをキャッシュメモリ内のコントローラ3000用領域3480に格納し、書き込み処理を完了する（ステップ909）。

【0038】プロセッサ3200は、プロセッサ4200に書き込み処理を要求するために、書き込みデータ格納アドレス、書き込みデータのキャッシュ上の格納アドレス、データ長及び処理情報等をデータ転送制御部

3300により共通メモリ領域3410、4410内のプロセッサ3200書き込み用メモリ内のプロセッサ4200への要求用エリアに2重に格納する。ここで、処理情報情報とは、書き込み処理が読み込み処理を判断することになった一定時間で、共通メモリ領域3410、4410の自プロセッサへの要求用エリアを4重に、他のプロセッサからの要求を認識する。

【0039】図6は、プロセッサ3200からの処理要求を受信したときのプロセッサ4200の処理を示すフローチャートである。前述の方法により、プロセッサ3200からの要求を認識（ステップ931）したプロセッサ4200は、プロセッサ3200書き込み用メモリ内のプロセッサ4200への要求用エリア内の処理情報3490と4490は、書き込み要求であることを認識する（ステップ932）。そして、プロセッサ4200は、プロセッサ3200書き込み用メモリ内のプロセッサ4200への要求用エリア内の書き込み論理アドレス、書き込みデータのキャッシュ上の格納先アドレス、データ長を取得し（ステップ936）、キャッシュ3400内の格納先アドレスからデータ長分の書き込みデータをプロセッサ4200用領域3490と4490に、その管理情報である書き込み論理アドレスとデータ長とを、2重に格納する（ステップ934）。そして、読了報告を共通メモリ領域3410、4410内のプロセッサ4200書き込み用メモリ内のプロセッサ3200からの要求に対する応答用エリアに設定することにより、プロセッサ3200に処理終了を通知する（ステップ935）。

【0040】プロセッサ3200は、プロセッサ4200に書き込み要求時、プロセッサ4200書き込み用メモリ内のプロセッサ3200からの要求に対する応答用エリアを監視することにより、プロセッサ4200の処理の終了を監視（ステップ910）して、その（図4参照）、この処理終了の通知を受けて、ホストコンピュータ1000に終了を報告する（ステップ905）。プロセッサ4200は、この後、図5に示すようにホスト1、〇処理とは異同期に、この書き込みデータのディスク装置7000への書き込み処理を行う。

【0041】図4において、ホストコンピュータ1000からの読み込み要求があったときは、プロセッサ3200は、上記書き込み要求受領後、処理相当論理ボリューム（LUN）への処理であることを認識し（ステップ902）後、処理後の判定を行う（ステップ903）。読み込み処理であることを認識すると、プロセッサ3200に読み込み要求論理アドレス、書き込みデータのキャッシュ上の格納先アドレス、データ長、処理情報等を共通メモリ領域3410、4410内のプロセッサ3200書き込みメモリ内のプロセッサ4200への要求用エリアに格納することにより、読了報告を共通メモリ領域3410、4410に書き込み要求時

- 6 - 2003 04 24 10 53

イのプロセッサには割り当てないことにより、キャッシュを有効に利用することができる。又、プロセッサの負荷に応じてダイナミックに変更することも可能である。ユーザの指定により分割を行うが、プロセッサの負荷に応じて変更を行うかの指示は、本実施例では、ホストコンピュータにより行いが、バネルといった装置を接続し、その入出力する部を取っても、むろん良い。

【0052】つぎに、コントローラのキャッシュの動的割当方式について、以下、説明する。まず、キャッシュの管理方式について、図9を用いて説明する。【0053】プロセッサ毎に持つデータ格納エリアは、セグメント983と呼ばれる管理単位に分割されている。セグメントは、セグメント毎にセグメント管理プロセッサ981（以下SGCBという。）をデータ管理情報内に持ち、セグメントを管理する情報とセグメントアドレスが格納されている。又、これらのSGCBは、そのセグメントの属性によって、データキュー980とクリーンキュー982という2つキューに分けられて格納されている。データキュー980は、ディスク装置に格納されたデータを格納しているセグメントのSGCBが格納されており、それ以外のSGCBは、クリーンキュー982に格納されている。

【0054】キャッシュの動的割当を実現するために、プロセッサ毎の負荷情報を共通メモリ領域に格納し、この負荷情報によって、例えば、キャッシュ内のクリーンSGCBを用いる。各プロセッサは、SGCBのクリーン、データ間のキュー選択処理に、この情報を使用する。プロセッサは、例えば、1分といった一定周期でこの情報を読み取り、キャッシュを共有しているプロセッサ内最も負荷の低いプロセッサのクリーンキューからキャッシュの高速（データ）に書き込むことができる。データロストを防ぐことができる。さらに、コントローラ毎にキャッシュメモリの制御を正常なコントローラに切り替える手段とコントローラ障害から復旧する手段とを設けることにより、システムを無停止で運用することが可能となる。

【0055】また、以上の実施例においては、2台のコントローラ間でキャッシュを共有し、各々、各コントローラのキャッシュに2重書きする例を示したが、キャッシュ領域がプロセッサ毎に分割されているため、各々のキャッシュを有形式化、多重書き方法は、任意の方法でも、同時に実現できる。

【0056】キャッシュ多重書きの例を図10に示す。（1）は、装置全体でキャッシュを共有し、2重書きする方法である。つまり、プロセッサ3200は共通メモリ領域3400、4400を用いて、プロセッサ4200は共通メモリ領域5400、6400を用いて、プロセッサ3200はキャッシュ6400、3400を用いて2重書きを行っている。

【0057】（2）は、装置全体でキャッシュを共有し、全キャッシュに多重書きする方法である。つまり、プロセッサ3200、4200、5200、6200は、それぞれキャッシュ3400、4400、5400、6400を用いて、2重書きを行っている。このケースにおいて、コントローラが障害となった場合は、キャッシュを共有しているプロセッサ間でもっとも負荷の低いプロセッサが、障害コントローラ相当論理ボリュームの処理を引き継ぐ。これらのケースにおいては、任意のプロセッサが障害コントローラに相当する論理ボリュームの処理を引き継げるように、ディスク側のデータバスを、装置内の全ディスク装置、全コントローラで共有のバスに接続しておく。もちろん、これらの多重書き方法を装置内で実現することも可能である。これらの多重書き方法の指定は、共通メモリ領域3410、4410に多重書き情報を格納し、各々のプロセッサ3200、4200、5200、6200が、この情報を元に、書き込みデータの転送方式をデータ転送制御部3300、4300、5300、6300に指示することにより実現できるとなる。

【0058】本発明によれば、コントローラ及びキャッシュメモリを2重化した記憶サブシステムにおいて、各コントローラにキャッシュメモリの一部及び論理ボリュームを割り当てることによりキャッシュメモリに対するコントローラ内のプロセッサ間の情報制御が簡便となる。複数プロセッサ化による応答性能向上を図ることができる。

【0059】また、複数のキャッシュへ多重書きすることにより、キャッシュ障害時には、多重書きしている他のキャッシュからディスクに書き込むことができる。データロストを防ぐことができる。さらに、コントローラ毎にキャッシュメモリの制御を正常なコントローラに切り替える手段とコントローラ障害から復旧する手段とを設けることにより、システムを無停止で運用することが可能となる。

【図9の簡易な説明】

【図1】本発明の構成を示す構成図である。

【図2】本発明の実施例であるコントローラの構成図である。

【図3】本発明の装置であるコントローラのキャッシュの構成を示す図である。

【図4】本発明の実施例によるコントローラの動作を示すフローチャートである。

【図5】本発明の実施例によるコントローラのキャッシュ内のデータをディスク装置に格納する動作を示すフローチャートである。

【図6】本発明の実施例による他のコントローラからの処理要求を受けたプロセッサ4200の動作を示すフローチャートである。

【図7】本発明の実施例による他のコントローラからの処理要求を受けたプロセッサ4200の動作を示すフローチャートである。

【図8】本発明の実施例による他のコントローラからの処理要求を受けたプロセッサ4200の動作を示すフローチャートである。

【図9】本発明の装置であるコントローラの構成図である。

【図10】本発明の装置であるコントローラの構成図である。

- 8 - 2003 04 24 10 53

【図7】本発明の実施例による他のコントローラへの追加を模したコントローラへの動作を示すフローチャートである。

【図8】本発明の実施例による追加が発生したコントローラの管理動作を示すフローチャートである。

【図9】本発明の実施例によるコントローラにおいて用いられるキャッシュの管理方式を示す図である。

【図10】本発明の実施例によるコントローラのキャッシュの構成を示す図である。

【符号の説明】

10:11:ホストコンピュータ

20:制御装置

30:40:コントローラ

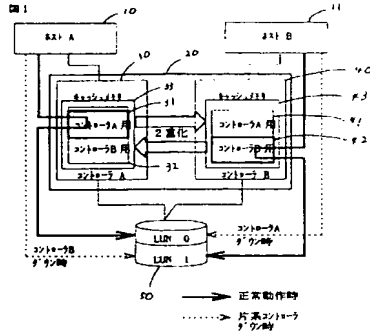
31:41:コントローラ用キャッシュメモリ

32:42:コントローラ用キャッシュメモリ

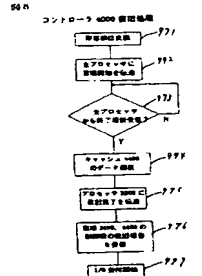
33:43:キャッシュメモリ

50:ディスク装置  
1000 1100 1200 1300:サブシステム  
2000:制御装置  
3000 1000 5000 6000:コントローラ  
3100 4100 5100 6100:ホスト  
F制御部  
3200 4200 5200 6200:マイクロプロセッサ  
10:10セツプ  
3300 4300 5300 6300:データ転送制御部  
3400 4400 5400 6400:キャッシュ  
3500 4500 5500 6500:DMV  
15:F制御部  
7000 7100:ディスク装置群

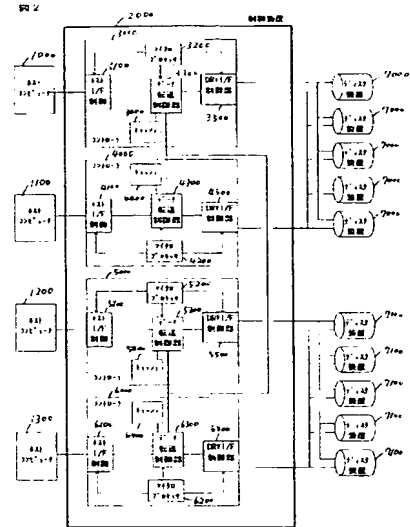
【図1】



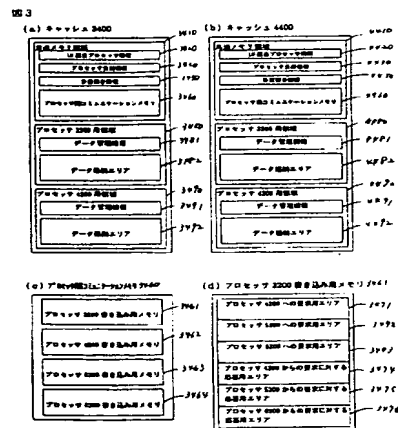
【図5】



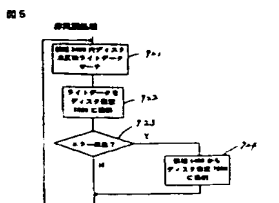
【図2】



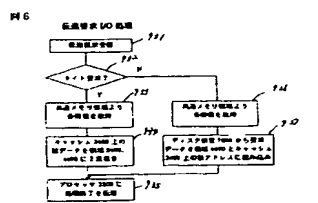
【図3】



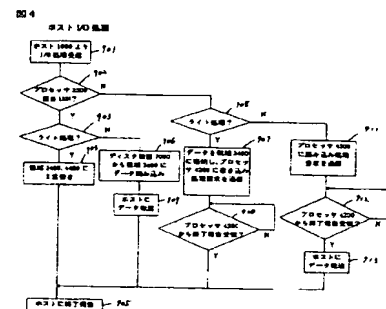
【図5】



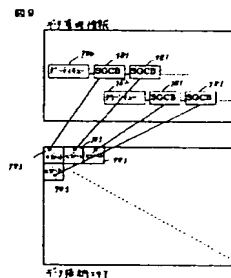
【図6】



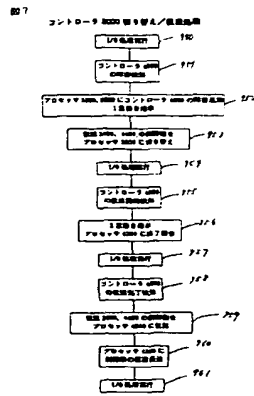
【図4】



【図9】



【圖 7】



(M 10)

